MEMORY HUB AND METHOD FOR MEMORY SYSTEM PERFORMANCE MONITORING

TECHNICAL FIELD

This invention relates to computer systems, and, more particularly, to a computer system having a memory hub coupling several memory devices to a processor or other memory access device.

BACKGROUND OF THE INVENTION

Computer systems use memory devices, such as dynamic random access memory ("DRAM") devices, to store data that are accessed by a processor. These memory devices are normally used as system memory in a computer system. In a typical computer system, the processor communicates with the system memory through a processor bus and a memory controller. The processor issues a memory request, which includes a memory command, such as a read command, and an address designating the location from which data or instructions are to be read. The memory controller uses the command and address to generate appropriate command signals as well as row and column addresses, which are applied to the system memory. In response to the commands and addresses, data are transferred between the system memory and the processor. The memory controller is often part of a system controller, which also includes bus bridge circuitry for coupling the processor bus to an expansion bus, such as a PCI bus.

Although the operating speed of memory devices has continuously increased, this increase in operating speed has not kept pace with increases in the operating speed of processors. Even slower has been the increase in operating speed of memory controllers coupling processors to memory devices. The relatively slow speed of memory controllers and memory devices limits the data bandwidth between the processor and the memory devices.

In addition to the limited bandwidth between processors and memory devices, the performance of computer systems is also limited by latency problems that

increase the time required to read data from system memory devices. More specifically, when a memory device read command is coupled to a system memory device, such as a synchronous DRAM ("SDRAM") device, the read data are output from the SDRAM device only after a delay of several clock periods. Therefore, although SDRAM devices can synchronously output burst data at a high data rate, the delay in initially providing the data can significantly slow the operating speed of a computer system using such SDRAM devices.

One approach to alleviating the memory latency problem is to use multiple memory devices coupled to the processor through a memory hub. In a memory hub architecture, a system controller or memory controller is coupled to several memory modules, each of which includes a memory hub coupled to several memory devices. The memory hub efficiently routes memory requests and responses between the controller and the memory devices. Computer systems employing this architecture can have a higher bandwidth because a processor can access one memory device while another memory device is responding to a prior memory access. For example, the processor can output write data to one of the memory devices in the system while another memory device in the system is preparing to provide read data to the processor.

Although computer systems using memory hubs may provide superior performance, they nevertheless often fail to operate at optimum speed for several reasons. For example, even though memory hubs can provide computer systems with a greater memory bandwidth, they still suffer from latency problems of the type described above. More specifically, although the processor may communicate with one memory device while another memory device is preparing to transfer data, it is sometimes necessary to receive data from one memory device before the data from another memory device can be used. In the event data must be received from one memory device before data received from another memory device can be used, the latency problem continues to slow the operating speed of such computer systems.

One technique that has been used to reduce latency in memory devices is to prefetch data, *i.e.*, read data from system memory before the data are requested by a program being executed. Generally the data that are to be prefetched are selected based

on a pattern of previously fetched data. The pattern may be as simple as a sequence of addresses from which data are fetched so that data can be fetched from subsequent addresses in the sequence before the data are needed by the program being executed. The pattern, which is known as a "stride," may, of course, be more complex.

Further, even though memory hubs can provide computer systems with a greater memory bandwidth, they still suffer from throughput problems. For example, before data can be read from a particular row of memory cells, digit lines in the array are typically precharged by equilibrating the digit lines in the array. The particular row is then opened by coupling the memory cells in the row to a digit line in respective columns. A respective sense amplifier coupled between the digit lines in each column then responds to a change in voltage corresponding to the data stored in respective memory cell. Once the row has been opened, data can be coupled from each column of the open row by coupling the digit lines to a data read path. Opening a row, also referred to as a page, therefore consumes a finite amount of time and places a limit on the memory throughput.

Finally, the optimal decision of whether or not to prefetch data (and which data to prefetch), as well as whether or not to precharge or open a row, and whether or not to cache accessed data, may change over time and vary as a function of an application being executed by a processor that is coupled to the memory hub.

Another potential problem with memory hub architectures relates to the use of a memory hub as a conduit for coupling memory requests and data through the memory hub to and from downstream memory modules. If the memory requests and data are not efficiently routed through the memory hub, the memory bandwidth of a memory system employing memory hubs can be severely limited.

All of the above-described issues can be addressed to some extent by configuring the memory module, including a memory hub mounted in the module, in different respects. However, before the configuration of the memory module can be optimized, it is necessary or desirable to analyze the performance of the memory hub so the areas in which performance is lacking can be determined. However, suitable

techniques to analyze the ongoing performance of memory systems used in processor-based system have not been developed.

There is therefore a need for a computer architecture that provides the advantages of a memory hub architecture that can also allow the performance of a memory system using the memory hub architecture to be determined so that the configuration of the system could be optimized.

SUMMARY OF THE INVENTION

According to one aspect of the invention, a memory module and method is provided including a plurality of memory devices and a memory hub. The memory hub contains a link interface, such as an optical input/output port, that receives memory requests for access to memory cells in at least one of the memory devices. The memory hub further contains a memory device interface coupled to the memory devices, the memory device interface being operable to couple memory requests to the memory devices for access to memory cells in at least one of the memory devices and to receive read data responsive to at least some of the memory requests. The memory hub further contains a performance counter coupled to the memory device interface and/or the link interface. The performance counter is operable to track at least one metric selected from the group consisting of page hit rate, number or percentage of prefetch hits, cache hit rate or percentage, read rate, number of read requests, write rate, number of write requests, rate or percentage of memory bus utilization, local hub request rate or number, and remote hub request rate or number.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a computer system according to one example of the invention in which a memory hub is included in each of a plurality of memory modules.

Figure 2 is a block diagram of a memory hub used in the computer system of Figure 1, which contains a performance monitor according to one example of the invention.

Figure 3 is a block diagram of a memory hub which contains performance counters according to one example of the invention.

DETAILED DESCRIPTION OF THE INVENTION

A computer system 100 according to one example of the invention is shown in Figure 1. The computer system 100 includes a processor 104 for performing various computing functions, such as executing specific software to perform specific calculations or tasks. The processor 104 includes a processor bus 106 that normally includes an address bus, a control bus, and a data bus. The processor bus 106 is typically coupled to cache memory 108, which, as previously mentioned, is usually static random access memory ("SRAM"). Finally, the processor bus 106 is coupled to a system controller 110, which is also sometimes referred to as a "North Bridge" or "memory controller."

The system controller 110 serves as a communications path to the processor 104 for a variety of other components. More specifically, the system controller 110 includes a graphics port that is typically coupled to a graphics controller 112, which is, in turn, coupled to a video terminal 114. The system controller 110 is also coupled to one or more input devices 118, such as a keyboard or a mouse, to allow an operator to interface with the computer system 100. Typically, the computer system 100 also includes one or more output devices 120, such as a printer, coupled to the processor 104 through the system controller 110. One or more data storage devices 124 are also typically coupled to the processor 104 through the system controller 110 to allow the processor 104 to store data or retrieve data from internal or external storage media (not shown). Examples of typical storage devices 124 include hard and floppy disks, tape cassettes, and compact disk read-only memories (CD-ROMs).

The system controller 110 is coupled to several memory modules 130a,b...n, which serve as system memory for the computer system 100. The memory modules 130 are preferably coupled to the system controller 110 through a high-speed link 134, which may be an optical or electrical communication path or some other type of communications path. In the event the high-speed link 134 is implemented as an

optical communication path, the optical communication path may be in the form of one or more optical fibers, for example. In such case, the system controller 110 and the memory modules will include an optical input/output port or separate input and output ports coupled to the optical communication path. The memory modules 130 are shown coupled to the system controller 110 in a point-to-point arrangement in which each segment of the high-speed link 134 is coupled between only two points. Therefore, all but the final memory module 130n is used as a conduit for memory requests and data coupled to and from downstream memory modules 130. However, it will be understood that other topologies may also be used. A switching topology may also be used in which the system controller 110 is selectively coupled to each of the memory modules 130 through a switch (not shown). Other topologies that may be used will be apparent to one skilled in the art.

Each of the memory modules 130 includes a memory hub 140 for controlling access to 32 memory devices 148, which, in the example illustrated in Figure 1, are synchronous dynamic random access memory ("SDRAM") devices. The memory hub 140 in all but the final memory module 130 also acts as a conduit for coupling memory commands to downstream memory hubs 140 and data to and from downstream memory hubs 140. However, a fewer or greater number of memory devices 148 may be used, and memory devices other than SDRAM devices may, of course, also be used. In the example illustrated in Figure 1, the memory hubs 140 communicate over 4 independent memory channels 149 over the high-speed link 134. In this example, although not shown in Figure 1, 4 memory hub controllers 128 are provided, each to receive data from one memory channel 149. A fewer or greater number of memory channels 149 may be used, however, in other examples. The memory hub 140 is coupled to each of the system memory devices 148 through a bus system 150, which normally includes a control bus, an address bus and a data bus.

A memory hub 200 according to an embodiment of the present invention is shown in Figure 2. The memory hub 200 can be substituted for the memory hub 140 of Figure 1. The memory hub 200 is shown in Figure 2 as being coupled to four memory devices 240a-d, which, in the present example are conventional SDRAM

devices. In an alternative embodiment, the memory hub 200 is coupled to four different banks of memory devices, rather than merely four different memory devices 240a-d, with each bank typically having a plurality of memory devices. However, for the purpose of providing an example, the present description will be with reference to the memory hub 200 coupled to the four memory devices 240a-d. It will be appreciated that the necessary modifications to the memory hub 200 to accommodate multiple banks of memory is within the knowledge of those ordinarily skilled in the art.

Further included in the memory hub 200 are link interfaces 210a-d and 212a-d for coupling the memory module on which the memory hub 200 is located to a first high speed data link 220 and a second high speed data link 222, respectively. The link interfaces 210a-d and 212a-d allow the memory hub 200 to be used as a conduit for memory requests and data to and from downstream memory modules 130. As previously discussed with respect to Figure 1, the high speed data links 220, 222 can be implemented using an optical or electrical communication path or some other type of communication path. The link interfaces 210a-d, 212a-d are conventional, and include circuitry used for transferring data, command, and address information to and from the high speed data links 220, 222. As is well known, such circuitry includes transmitter and receiver logic known in the art. It will be appreciated that those ordinarily skilled in the art have sufficient understanding to modify the link interfaces 210a-d, 212a-d to be used with specific types of communication paths, and that such modifications to the link interfaces 210a-d, 212a-d can be made without departing from the scope of the present invention. For example, in the event the high-speed data link 220, 222 is implemented using an optical communications path, the link interfaces 210a-d, 212a-d will include an optical input/output port that can convert optical signals coupled through the optical communications path into electrical signals.

The link interfaces 210a-d, 212a-d are coupled to a switch 260 through a plurality of bus and signal lines, represented by busses 214. The busses 214 are conventional, and include a write data bus and a read data bus, although a single bi-directional data bus may alternatively be provided to couple data in both directions through the link interfaces 210a-d, 212a-d. It will be appreciated by those ordinarily

skilled in the art that the busses 214 are provided by way of example, and that the busses 214 may include fewer or greater signal lines, such as further including a request line and a snoop line, which can be used for maintaining cache coherency.

The link interfaces 210a-d, 212a-d include circuitry that allow the memory hub 200 to be connected in the system memory in a point-to-point configuration, as previously explained.This type of interconnection provides better signal coupling between the processor 104 and the memory hub 200 for several reasons, including relatively low capacitance, relatively few line discontinuities to reflect signals and relatively short signal paths. However, the link interfaces 210a-d and 212a-d could also be used to allow coupling to the memory hubs 200 in a variety of other configurations.

The switch 260 is further coupled to four memory interfaces 270a-d which are, in turn, coupled to the system memory devices 240a-d, respectively. By providing a separate and independent memory interface 270a-d for each system memory device 240a-d, respectively, the memory hub 200 avoids bus or memory bank conflicts that typically occur with single channel memory architectures. The switch 260 is coupled to each memory interface through a plurality of bus and signal lines, represented by busses 274. The busses 274 include a write data bus, a read data bus, and a request line. However, it will be understood that a single bi-directional data bus may alternatively be used instead of a separate write data bus and read data bus. Moreover, the busses 274 can include a greater or lesser number of signal lines than those previously described.

In an embodiment of the present invention, each memory interface 270a-d is specially adapted to the system memory devices 240a-d to which it is coupled. More specifically, each memory interface 270a-d is specially adapted to provide and receive the specific signals received and generated, respectively, by the system memory device 240a-d to which it is coupled. Also, the memory interfaces 270a-d are capable of operating with system memory devices 240a-d operating at different clock frequencies. As a result, the memory interfaces 270a-d isolate the processor 104 from changes that may occur at the interface between the memory hub 230 and memory

devices 240a-d coupled to the memory hub 200, and it provides a more controlled environment to which the memory devices 240a-d may interface.

The switch 260 coupling the link interfaces 210a-d, 212a-d and the memory interfaces 270a-d can be any of a variety of conventional or hereinafter developed switches. For example, the switch 260 may be a cross-bar switch that can simultaneously couple link interfaces 210a-d, 212a-d and the memory interfaces 270a-d to each other in a variety of arrangements. The switch 260 can also be a set of multiplexers that do not provide the same level of connectivity as a cross-bar switch but nevertheless can couple the some or all of the link interfaces 210a-d, 212a-d to each of the memory interfaces 270a-d. The switch 260 may also include arbitration logic (not shown) to determine which memory accesses should receive priority over other memory accesses. Bus arbitration performing this function is well known to one skilled in the art.

With further reference to Figure 2, each of the memory interfaces 270a-d includes a respective memory controller 280, a respective write buffer 282, and a respective cache memory unit 284. The memory controller 280 performs the same functions as a conventional memory controller by providing control, address and data signals to the system memory device 240a-d to which it is coupled and receiving data signals from the system memory device 240a-d to which it is coupled. The write buffer 282 and the cache memory unit 284 include the normal components of a buffer and cache memory, including a tag memory, a data memory, a comparator, and the like, as is well known in the art. The memory devices used in the write buffer 282 and the cache memory unit 284 may be either DRAM devices, static random access memory ("SRAM") devices, other types of memory devices, or a combination of all three. Furthermore, any or all of these memory devices as well as the other components used in the cache memory unit 284 may be either embedded or stand-alone devices.

The write buffer 282 in each memory interface 270a-d is used to store write requests while a read request is being serviced. In a such a system, the processor 104 can issue a write request to a system memory device 240a-d even if the memory device to which the write request is directed is busy servicing a prior write or read

request. Using this approach, memory requests can be serviced out of order since an earlier write request can be stored in the write buffer 282 while a subsequent read request is being serviced. The ability to buffer write requests to allow a read request to be serviced can greatly reduce memory read latency since read requests can be given first priority regardless of their chronological order. For example, a series of write requests interspersed with read requests can be stored in the write buffer 282 to allow the read requests to be serviced in a pipelined manner followed by servicing the stored write requests in a pipelined manner. As a result, lengthy settling times between coupling write request to the memory devices 270a-d and subsequently coupling read request to the memory devices 270a-d for alternating write and read requests can be avoided.

The use of the cache memory unit 284 in each memory interface 270a-d allows the processor 104 to receive data responsive to a read command directed to a respective system memory device 240a-d without waiting for the memory device 240a-d to provide such data in the event that the data was recently read from or written to that memory device 240a-d. The cache memory unit 284 thus reduces the read latency of the system memory devices 240a-d to maximize the memory bandwidth of the computer system. Similarly, the processor 104 can store write data in the cache memory unit 284 and then perform other functions while the memory controller 280 in the same memory interface 270a-d transfers the write data from the cache memory unit 284 to the system memory device 240a-d to which it is coupled.

Further included in the memory hub 200 is a performance monitor 290 coupled to the switch 260 through a diagnostic bus 292. The performance monitor 290 monitors the performance of the memory hub 200 through the switch 260. For example, the performance monitor 290 can keep track of the number of cache hits, memory page hits or prefetch hit occurring in the memory hub 200. The performance monitor 290 can also monitor the coupling of memory requests and data through the memory hub 200 via the link interfaces 210a-d, 212a-d and the switch 260 to determine how busy the hub 200 is and whether it is coupling memory requests and data efficiently and without excessive delay. The performance monitor 290 is further coupled to a

maintenance bus 296, such as a System Management Bus (SMBus) or a maintenance bus according to the Joint Test Action Group (JTAG) and IEEE 1149.1 standards. Both the SMBus and JTAG standards are well known by those ordinarily skilled in the art. Generally, the maintenance bus 296 provides a user access to the performance statistics tracked by the performance monitor 290. It will be appreciated that the maintenance bus 296 can be modified from conventional bus standards without departing from the scope of the present invention. It will be further appreciated that the performance statistics can be coupled from the performance monitor 290 by other means.

Further included in the memory hub 200 is a DMA engine 286 coupled to the switch 260 through a bus 288. The DMA engine 286 enables the memory hub 200 to move blocks of data from one location in the system memory to another location in the system memory without intervention from the processor 104. The bus 288 includes a plurality of conventional bus lines and signal lines, such as address, control, data busses, and the like, for handling data transfers in the system memory. Conventional DMA operations well known by those ordinarily skilled in the art can be implemented by the DMA engine 286. The DMA engine 286 is able to read a link list in the system memory to execute the DMA memory operations without processor intervention, thus, freeing the processor 104 and the bandwidth limited system bus from executing the memory operations. The DMA engine 286 can also include circuitry to accommodate DMA operations on multiple channels, for example, for each of the system memory devices 240a-d. Such multiple channel DMA engines are well known in the art and can be implemented using conventional technologies.

The performance monitor 290 is preferably an embedded circuit in the memory hub 200. However, including a separate performance monitor coupled to the memory hub 200 is also possible.

As described above, one approach to reducing latency in memory devices is to prefetch data. Accordingly, the memory hub 200 further includes a prefetch system 295 including a prefetch buffer 298. Briefly, the prefetch system 295 in the memory hub 200 anticipates which data will be needed during execution of a program, and then prefetches those data and stores them in one or more buffers, such as a prefetch buffer

298, that are part of the prefetch system 295. The prefetch system 295 includes several prefetch buffers, including the prefetch buffer 298, the number of which can be made variable depending upon operating conditions, as explained in the aforementioned patent application. Briefly, the prefetch buffers receive prefetched data from the memory device interface 270c in Figure 2. The data are stored in the prefetch buffers so that they will be available for a subsequent memory access. The data are then coupled to the link interface 212d. Although one prefetch system is shown coupled to the memory device interface 270c and the link interface 212d in Figure 2, it is to be understood that in some embodiments the prefetch system 295 may be coupled to a plurality of link interfaces and/or a plurality of memory device interfaces. Further, in some embodiments, a plurality of prefetch systems may be provided in communication with one or a plurality of link interfaces and/or memory device interfaces.

Another embodiment of a portion of a memory hub that obtains performance data is shown in Figure 3. In the example illustrated in Figure 3, at least one performance counter 300 is provided in communication with a memory controller 302. The performance counter 300 is further in communication with a prefetch buffer 306, a cache 308, links 310 and 314, and a maintenance bus 318. It is to be understood that in some examples of the invention, the performance counter 300 may not be in communication with one or more of the components shown in Figure 3. Further, in still other examples one or more performance counters are in communication with other elements of a memory hub not shown in Figure 3.

The performance counters 300 track one or more metrics associated with memory access and/or performance, including for example, page hit rate, number or percentage of prefetch hits, cache hit rate or percentage, read rate, number of read requests, write rate, number of write requests, rate or percentage of memory bus utilization, local hub request rate or number, and remote hub request rate or number, in one example of the invention. The performance counters 300 also monitors the coupling of memory requests and data through the memory hub to determine how busy the hub is and whether it is coupling memory requests and data efficiently and without excessive delay. It is to be understood that the performance counter 300 could monitor

other performance characteristics, depending on the configuration of the memory hub and the components with which the counter is in communication. In either case, the performance counter 300 tracks performance characteristics, and preferably communicates those characteristics from the memory modules 130 so that they can be examined. For example, the data indicative of the performance characteristics can be coupled through the maintenance bus 318. The maintenance bus 318 can provides a user with access to the performance counters 300 to assess the performance of the computer system. For example, performance characteristics can be downloaded to a separate PC host via the maintenance bus 318. Other means of coupling and/or using the performance characteristics will be apparent to one skilled in the art.

From the foregoing it will be appreciated that, although specific embodiments of the invention have been described herein for purposes of illustration, various modifications may be made without deviating from the spirit and scope of the invention. Accordingly, the invention is not limited except as by the appended claims.